Mitigating Spam Using Spatio-Temporal Reputation

Andrew G. West[*] Adam J. Aviv[†]
Jian Chang[‡] Insup Lee[**]

[*]University of Pennsylvania
[†]University of Pennsylvania
[‡]University of Pennsylvania
[**]University of Pennsylvania, lee@cis.upenn.edu

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-10-04.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE **Mitigating Spam Using Spatio-Temporal Reputation**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Dept. of Computer and Information Science,University of Pennsylvania ,Philadelphia,PA**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**21**	

Mitigating Spam Using Spatio-Temporal Reputation[*]

Andrew G. West Adam J. Aviv Jian Chang Insup Lee
Dept. of Computer and Information Science - University of Pennsylvania - Philadelphia, PA
{westand, aviv, jianchan, lee}@cis.upenn.edu

Abstract

In this paper we present **Pre**ventive **S**patio-**T**emporal **A**ggregation (PRESTA), a reputation model that combines spatial and temporal features to produce values that are behavior predictive and useful in partial-knowledge situations. To evaluate its effectiveness, we applied PRESTA in the domain of spam detection. Studying the temporal properties of IP blacklists, we found that 25% of IP addresses once listed on a blacklist were re-listed within 10 days. Further, during our evaluation period over 45% of IPs de-listed were re-listed. By using the IP address assignment hierarchy to define spatial groupings and leveraging these temporal statistics, PRESTA produces reputation values that correctly classify up to 50% of spam email not identified by blacklists alone, while maintaining low false-positive rates. When used in conjunction with blacklists, an average of 93% of spam emails are identified, and we find the system is consistent in maintaining this blockage rate even during periods of decreased blacklist performance. PRESTA spam filtering can be employed as an intermediate filter (perhaps in-network) prior to context-based analysis. Further, our spam detection system is scalable; computation can occur in near real-time and over 500,000 emails can be scored an hour.

1 Introduction

Roughly 90% of the total volume of email on the Internet is considered spam [5], and IP-based blacklisting has become a standard tool in fighting such influxes. The vast majority of spam originates from *botnets*, centrally controlled organizations of infected hosts. Spammers can vary which hosts within a botnet are sending spam, migrating as hosts become blacklisted. As a result, some 20% of spam emails received at a large spam trap in 2006 were not listed on any blacklist [25].

Despite the dynamic nature of spamming IP addresses, they still exhibit interesting spatial properties. In particular, the majority of such IPs are clustered throughout the address space [25]; Spammers tend to be spatially found "near" other spammers (*e.g.*, within the same subnet or Autonomous System (AS)). This phenomenon was recently reinforced when in December of 2008 the McColo ISP was shutdown by authorities and spam rates dropped by nearly 15% [5, 21]. Further, previous studies have demonstrated that the AS-level membership of spamming IPs can be used to effectively classify spam [15].

Temporal properties of spam originating IPs, however, have *not* been studied as a measure of sender reputation. The history of a single IP's entry and exit from a blacklist is significant. More crucially, the listing and de-listing patterns of *spatially related* IP addresses *over time* is behavior predictive. For example, suppose a subnet has a dense blacklist history. If half of the subnet is currently or was recently blacklisted, it would be prudent to view the other half suspiciously. In our analysis, we found that previous blacklistings are predictive of future listings. More than 25% of IPs once listed were re-listed within 10 days, and 45% were re-listed during our experiment period. Given these statistics, it becomes clear that the usage of historical data, as opposed to the static view blacklists provide, may be most helpful when valuating unknown entities.

[*] This research was supported in part by ONR MURI N00014-07-1-0907. POC: Insup Lee, lee@cis.upenn.edu

Further, we expand previous work regarding spatial groupings of spam originating IP addresses by using knowledge of the IP assignment hierarchy. We found that calculating reputation at multiple levels provides greater insight into unknown quantities. For example, the clustering of spamming IPs within an AS are distributed non-uniformly, just as malignant ASes are found non-uniformly across the address space. We find a combination of both fine and broad groupings best correlate negative feedback.

We have developed a novel reputation model, PRESTA (**Pre**ventive **S**patio-**T**emporal **A**ggregation), that computes values that are behavior predictive of unknown entities in partial-knowledge situations where entity-specific data may be incomplete. PRESTA aggregates a history of negative feedback from a database (*i.e.*, a blacklist) into *reputations* for entities based on their individual behavior and that of their spatial groupings. PRESTA requires only that, (1) there exists a grouping function to define finite sets of participants, and (2) there is observable negative feedback to construct a behavior history. For each spatial grouping, a separate reputation value is produced and machine learning can be applied to tune the combination to generate a higher-order reputation. This value can incorporate as many spatial and temporal dimensions as necessary to best represent the entities being valuated.

Spam is a motivating model for PRESTA, and we use it to analyze five months of mail logs provided by our university mail system. We do this by simulating a production email server implementing PRESTA, complete with caching and periodic re-training. Empirical results show that PRESTA can identify up to 50% of spam emails not caught by blacklists while maintaining low false-positive rates. When used in combination, PRESTA and classic blacklists are able to *consistently* identify 93% of spam without text-based analysis. Moreover, these blockage rates are *stable*, remaining steady-state even as the performance of the underlying blacklist suffers. Further, this implementation is *scalable*, scoring at latencies unnoticeable to an end user, while valuating over 500,000 emails/hour. We do not propose that PRESTA can replace context-based filtering, but instead may be used as an intermediate filter (perhaps in-network), reducing the number of emails that require complex text-processing and the exposure of private email content.

However, PRESTA defines a general model for spatio-temporal reputation which is broader than spam and IP blacklists alone. Any application that meets the two usage criteria of PRESTA can benefit. Preliminary work [32] has already shown PRESTA reputation values helpful in detecting vandalism on Wikipedia. More generally, PRESTA may be applicable to the entire class of dynamic trust management systems [10, 31], characterized by the need for decision-making in the presence of uncertainty. The Wikipedia use-case and other potential applications are given greater attention in Sec. 8.

2 Related Work

Previous work within the spam detection community has focused on using network-level properties to distinguish legitimate and malicious mail. Unlike content-based filters that employ Bayesian quantifiers [28], these techniques leverage the reputation of the sender – often the IP address of the connecting mail server.

IP[1] blacklists [3], such as Spamhaus [7], collate IP addresses of known spamming hosts based on feedback from varied institutions (large email providers). Over time, IP addresses may be listed, de-listed, and re-listed. It is this dynamic nature of blacklists that PRESTA leverages.

Recent work suggests that blacklists are too latent in their listing of new spammers [24], and that 10% of spamming IPs have never been seen previously [27]. Even so, studies have shown that the spamming IPs are distributed non-uniformly throughout the address space. Further, a large percentage of spam originates from small regions of the address space [25]. Generally, spammers are found adjacent to other spammers, and this principle was shown to be an effective metric in detecting *new* spammers by Hao *et al.* [15] in their

[1] IP blacklists are sometimes referred to as *DNS blacklists*, due to the frequent use of the DNS protocol in performing lookups (DNS lookup is used by mail servers that choose not to store a local copy of the blacklist). These lists contain only IPs.

development of the SNARE system. In particular, Hao *et al.* showed the AS-membership of an IP address to be relevant. We build upon this research, grouping IPs at multiple levels of the address assignment hierarchy.

In combination with spatial measurement, SNARE also utilizes *simple* temporal metrics to perform spam filtering (for example, the time-of-day an email was sent), and apply a lightweight form of aggregation (*i.e.*, mean and variance) over such features to detect abnormal patterns. In contrast, PRESTA's temporal aspect is more complex, aggregating time-decayed behavioral observations that encode *months* of *detailed* history. Indeed, [15] identifies many valid measures of spamming behavior, but is ultimately incapable of Internet-wide scalability given their reliance on high-dimensional learning. PRESTA spam detection computes over a single feature, IP address (and groups thereof), and is effective while realizing wide-scalability.

Outside of academia, several commercial services claim the use of techniques similar to those proposed herein. For example, Symantec [30] allows one to look-up "IP reputation", a feature used in its security software. For the public-facing service, the response to queries is binary – although we found it correlated well with the values PRESTA calculates. In addition, the SenderBase [16] system uses spatial data to build reputations. Whatever the internals of such products, we believe the PRESTA system makes a contribution by introducing this technology into a non-proprietary setting.

PRESTA can also be examined in the context of general-purpose reputation systems/logics, such as EigenTrust [20] or TNA-SL [18]. The primary distinguishing factor between these systems and PRESTA is the nature of feedback. Conventional algorithms usually aggregate both positive and negative feedback, where feedbacks are perpetually retained and associated to a single *discrete* time-stamp. In contrast, PRESTA considers only negative feedbacks, and functions best with *expiring feedback*, where a behavioral observation is valid for some finite duration and then discarded. A transformation can be applied so PRESTA can operate using more conventional feedback, a topic discussed further in Sec. 3.2.

Finally, most reputation systems are designed from a distributed perspective, concentrating on how untrusted parties can identify reliable partners in a network setting. In our application, the blacklist provider renders a singular and fully-trusted perspective. Moreover, while distributed reputation-network approaches to spam-filtering [9, 13] exist, these focus on sharing classifications rather than the discovery of new ones.

3 PRESTA **Reputation Model**

Summarily, the reputation model begins by mapping negative feedback to some identifier (*i.e.*, a user). Poor reputations correspond to those who have recently committed bad behavior (resulting in a negative feedback). In the absence of bad behavior, reputations improve over time. Critically, user-level reputations may also be combined according to the spatial relationships of the associated entities.

We consider a database where an entity is considered *active* when it receives negative feedback (becomes listed) and *inactive* when that negative feedback expires (becomes de-listed). Further, one is able to query this database to see an entity's listing history. When an entity has recently been assigned a negative feedback, the penalty associated with the listing is a heavy one. However, the more time that passes, the less weight a prior listing carries. When an entity is re-listed (re-enters the database after removal), the weight of the previous negative activity is still applied, but there is now compounding evidence against the entity being valuated. These characteristics are well represented in PRESTA where each listing, re-listing, and de-listing carries a weight based on its temporal relevance, realized via a decay function.

Spatial properties are considered based on grouping functions which take an entity as input and return all entities that are a member of the input's group. We expect more than one grouping function will be given, and an entity will populate multiple groups – one for each grouping function. This is advantageous; in the absence of entity-specific data, spatial (group) data is helpful in characterizing an entity's expected behavior. A reputation can then be computed at each grouping and combined according to application-specific criteria.

3.1 Reputation Computation

The goal of the reputation computation is to produce a quantified value that captures both the spatial and temporal properties of the entity being valuated. Spatially, the size of the grouping must be considered, and temporally, the history of negative feedback must be weighted proportional to its spatial proximity.

For example, suppose we are valuating the group-level reputation of an entity. If only a small portion of members have recent listing activity, then the reputation should be relatively high. Similarly, if a large portion of group members have a rich listing history, but in the distant past, the reputation should still be high. However, if many members have recent listing activity, the reputation should be relatively low.

To capture these properties, three functions are required – two temporal and one spatial:

- $\boldsymbol{hist(\alpha, G, H)}$ is a temporal function returning a list of pairs, (t_{in}, t_{out}), representing all entries/exits from the feedback history, H, according to the grouping of entity α by grouping function G. The values t_{in} and t_{out} are time-stamps bounding the listing. Active listings return $(t_{in}, \perp)$.

- $\boldsymbol{decay(t_{out}, h)}$ is a temporal function that exponentially decays input times using a half-life h, and it takes the form $2^{-\Delta t/h}$ where $\Delta t = t_{now} - t_{out}$ are of the same unit as h. It returns a value in the range $[0, 1]$, and for consistency, $decay(\perp, h) = 1$.

- $\boldsymbol{size(\alpha, G, t)}$ is a spatial function returning the magnitude, at time t, of the grouping defined by G, of which α is/was a member. If G defines multiple groupings for an entity α, only the magnitude of one grouping is returned. The choice of which group is application specific.

We now define the raw reputation function as follows:

$$\boldsymbol{raw\_rep(\alpha, G, H)} = \sum_{\substack{(t_{in}, t_{out}) \in \\ hist(\alpha, G, H)}} \frac{decay(t_{out}, h)}{size(\alpha, G, t_{in})} \tag{1}$$

This computation captures precisely the spatio-temporal properties required by PRESTA. Temporally, the entry/exit history of the database is captured at each summation via the $hist()$ function, and events occurring recently are more strongly weighted via the $decay()$ function. Spatially, grouping function G defines the group membership, and each summation is normalized by the group size.

When two or more grouping functions are defined over the entities, multiple computations of $raw\_rep()$ are performed. Each value encodes the reputation of an entity when considered within a different spatial context. How to best combine reputation is an application specific decision, and for our spam application, machine learning techniques are used (see Sec. 5.7).

The values returned by $raw\_rep()$ are strictly comparable for all spatial groupings defined by G and the history H. High values correspond to less reputable entities and vice-versa. However, it is more typical for reputation systems [18, 20] to normalize values onto the interval $[0, 1]$ where lower values correspond to low reputation and vice-versa. Ultimately, our machine learning technique does not require normalized values. Such values do, however, enable our model to be consistent with other reputation systems and provide an absolute interpretation that permits manually-authored policies (*e.g.*, allow access where reputation > 0.8).

Normalization requires knowledge of an upper bound on the values returned by $raw\_rep()$. This cannot be generally defined when the de-listing policy is non-regular. However, if it is known that listings expire from the negative feedback database after a fixed duration d, and that listings are non-overlapping, then it is possible to compute an upper bound. Such a bound can be found by considering an entity who is as bad as possible; one that is re-listed immediately after every de-listing, and thus, is always active in the feedback database. Considering a grouping of size 1, the $raw\_rep()$ computation reduces to a geometric sequence:

$$\texttt{MAX\_REP} = 1 + \frac{1}{1 - 2^{-d/h}} \tag{2}$$

Similarly, the same worst case reputation occurs for groups of larger size, however, instead of a single entity acting as a bad as possible, the entire group is simultaneously re-listed immediately following each de-listing. We can now define a normalized reputation as:

$$\boldsymbol{rep(\alpha, G, H)} = 1 - \left(\frac{raw\_rep(\alpha, G, H)}{\texttt{MAX\_REP}} \right) \tag{3}$$

Clearly, small d produce larger MAX_REP values. Thus, a precise value for d need not be known – only a lower-bound. One may ask, "Why not choose an arbitrarily small d (*i.e.*, 1 second) as the lower bound?" Such a small lower bound will increase the normalization factor and the distance between what can be considered good and bad reputation will become arbitrarily small. The best choice for easy differentiation is the greatest lower-bound of the listing interval. We found that a good estimate for d is sufficient, at least in the domain of spam detection, because the worst-case reputation is unlikely to be realized.

The reputation computation defined herein can be further specialized depending on the entities being valuated or the nature of the negative feedback database. For example, one can eliminate all spatial relevance by using grouping functions that define groups of size 1. Or, one can eliminate all temporal calculation by defining the return of $decay()$ as a constant (C). Both such usages are employed in spam detection; the former due to dynamism in IP address assignment, and the latter due to properties of the blacklist in question. Note that when $decay(t_{out}, h) = C$, $\texttt{MAX\_REP} = decay(\perp, h) + C$.

3.2 Reputation Database

The reputation database, H, depends on the nature of feedback available. PreSTA is most adept at handling *expiring* feedback like that present in IP blacklists. By definition, an expiring feedback occurs when an entity is active (listed) in the database before being removed (de-listed) after some finite duration. In this case, H is a record of the entries/exits of listings such that the active database can be reproduced at any point in time.

Feedback can also be *discrete*, where negative feedbacks are associated with a single time-stamp but stored for perpetuity. This is the model most frequently seen in general-purpose reputation management systems [18, 20]. In such cases, $hist()$ always returns pairs of the form $(t_{in}, \perp)$, and thus the associated listings do not decay. A discrete database can be transformed into a compatible H by setting an artificial timeout to allow for decay (*e.g.*, $(t_{in}, t_{in} + x)$ where x is the timeout).

Overlapping listings (*i.e.*, an entity having more than one active listing at a time) are not a concern for our spam implementation. It is the case for IP blacklists that an entity is actively listed at most once, and such guarantees are helpful in normalizing reputation values. Nonetheless, overlapping listings can be handled by PreSTA, provided some pre-processing is performed over the feedbacks.

4 Spam Detection Setup

In this section, we describe the application of the PreSTA model for the purpose of spam detection. Two properties of spam and IP blacklists are well leveraged by PreSTA. First, spammers are generally found "near" other spammers. Their identifiers, IP addresses, are easily grouped spatially due to the hierarchical nature of IP address assignment. Second, blacklists are a rich source of easily accessible temporal data.

It should be emphasized that IP blacklists are not a prerequisite to using PreSTA for spam detection. Any manner of negative feedback associating spamming and IP addresses is sufficient. IP blacklists, however, are a well-regarded and generally trusted source of negative feedback. They are centrally maintained and reputation computed upon them can be seen as a good global quantifier. However, IP blacklists do have weaknesses, and readers should take care not to associate these flaws to the PreSTA model.

4.1 Data Sources

Blacklist: To collect blacklist data, we subscribe to a popular blacklist-provider, Spamhaus [7]. The arrival and exit of IP addresses listed on three Spamhaus blacklists (updated at thirty-minute intervals) were recorded for the duration of the experiment[2]:

- Policy Block List (PBL): Listing of dynamic IP addresses (*e.g.*, those provided by large ISPs such as Comcast or Verizon), from which mail should never originate, on principle.
- Spamhaus Block List (SBL): Manually-maintained listing of IPs of known spammers and spam organizations. Typically these are IPs mapping to dedicated spam servers.
- Exploits Block List (XBL): Automated listing of IPs caught spamming; usually open proxies or machines that have been compromised by a bot-net.

As the latter two blacklists contain IP addresses known to have participated in spamming, only these are used to build reputation. The PBL is a preventative measure (though we do use its entries when examining blacklist performance). The mechanism by which a blacklist entry occurs, be it accurate or otherwise, is beyond the scope of this work. PreSTA considers all negative feedback equally, and as such, is not dependent on the means by which an IP becomes blacklisted. Some Spamhaus blacklists (PBL and SBL) list IP-prefixes (blocks) as opposed to individual IPs, but this is no different than listing each IP independently.

Removal from the blacklist takes two forms: manual de-listing and timed-expiration. Given its rigorous human maintenance, the SBL follows the former format. The XBL, on the other hand, defaults to a more automated time-to-live style. Empirical evidence shows the bulk of such listings expire 5-days after their appearance (see Fig. 1). However, in the case a blacklisted party can demonstrate its innocence or show the spam-generating exploit has been patched, manual removal is also an option for the XBL. Manual de-listings can complicate the calculation of MAX_REP, but as we will show, worst case spamming behaviors are rarely realized, permitting strong normalization.

AS Mappings: For the purpose of mapping an IP address to the Autonomous System(s) that *homes* or *originates* it, we use the reports generated by CAIDA [2]. These are compiled from Route Views [8] data and are essentially a snapshot of the BGP routing table.

E-mail Set: For testing purposes, we procured approximately 31 million email headers collected at the University of Pennsylvania engineering email servers between 8/1/2009 and 12/31/2009. The mail servers host over 6,100 accounts, of which approximately 5,500 serve human-users, while the remaining are for various administrative and school uses (*i.e.*, courses, aliases, lists, *etc.*). Pertinent information included only the time-stamp of receipt and the connecting email server's IP address [14].

A considerable number of emails (2.8 million) in our data-set were both sent and received within the university network. Such emails were not considered in our analysis. Many intra-network messages are the result of list-serves and aliasing, and by removing them, only externally arriving emails are considered. Our working set is further reduced to 6.1 million emails when analysis is conducted "above the blacklist" – focusing only on those mails passing IP blacklists.

To categorize email in our data-set as either spam or ham (not spam), we were provided a Proofpoint [6] score (in addition to the aforementioned headers). Proofpoint is a commercial spam detection service employed by the University whose detection methods are known to include proprietary filtering and Bayesian

[2] Although blacklist data is pulled every 30 minutes, a time-stamp is provided to identify precisely when a new IP is listed. However, no time-stamp is available for de-listing, and the timing of this event is inferred.

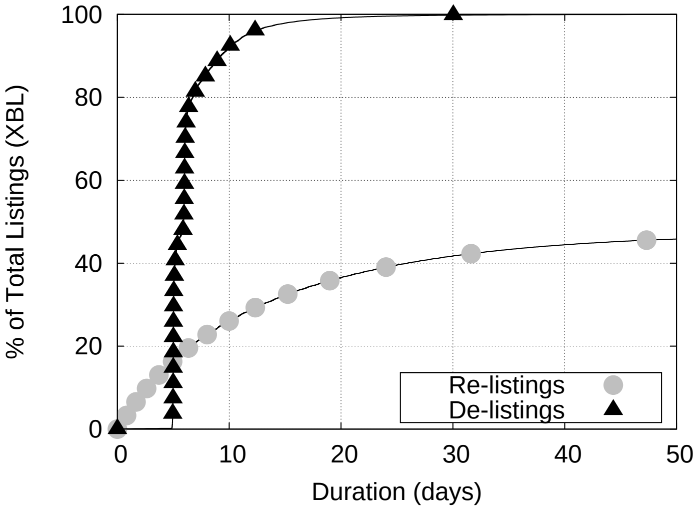

Figure 1: XBL Listing Duration & Re-listing Rates

content analysis [28] similar to that employed by SpamAssassin [1], an open source alternative. Proofpoint claims 99.8% accuracy with a low false-positive rate. Given no other consistent scoring metric and a lack of access to the original email bodies, we use the Proofpoint score as the control classifier for our analysis.

A fair question is then, "If text-based analysis is so accurate, what benefit is PRESTA?" The gains are three-fold: (1) Text-based analysis exposes private email to third party sources (often such services are web-based, as opposed to local software). (2) It is far easier for spammers to change email bodies than their mail-server's IP address. And, (3) PRESTA can be implemented as a lightweight pre-processor (perhaps in-network) to such text-based filtering, reducing the overhead of computationally expensive analysis.

4.2 Temporal Properties

PRESTA leverages the temporal properties of IP blacklists by aggregating the de-listing and re-listing rates of blacklist entries. Fig. 1 displays our analysis of those two statistics. Of IP-addresses de-listed during our experiment, 26% were re-listed within 10 days. Overall, 47% of such IPs were re-listed within 10 weeks, and it is precisely such statistics that motivate PRESTA's use of temporal data

Given that IP addresses are frequently re-listed, we examined the rate at which de-listing occurs. Empirical analysis shows that 80% of XBL entries were de-listed at, or very close to, 5 days after their entry (see Fig. 1). Even so, this 5-day interval is not fixed. Despite a non-exact expiration, `MAX_REP` is well computed using $d = 5$ (days) as a lower-bound. Raw reputation values rarely exceeded the calculated `MAX_REP` (less than 0.01% of the time). Moreover, the worst-case reputation is achieved when the time to re-listing is effectively zero. The shortest observed re-listing interval was approximately 6 minutes, and re-listings intervals on this order of duration are extremely rare.

Manually maintained, the SBL has no consistent listing length, and computing `MAX_REP` cannot be performed using the same analysis. Instead, the manual maintenance of the blacklist can be used as a factor in reputation. For an IP to be de-listed, it must be verified as a non-spamming address. Thus, there is no reason to decay entries as they exit the list[3]. That is, $\forall t_{out},\ decay(t_{out}) = 0$, but $decay(\perp)$ is still 1. In such circumstances, the `MAX_REP` value for such IPs is computed as 1 (*i.e.*, the IP address is currently listed).

[3] A reader's intuition may be, "If SBL entries are not decayed and the decay output simply echoes the SBL status, why is the SBL needed for reputation?" The answer is spatial: SBL and XBL listed IPs may reside inside the same spatial grouping, and SBL listings are representative of malicious behavior which should lower the reputation for that grouping.

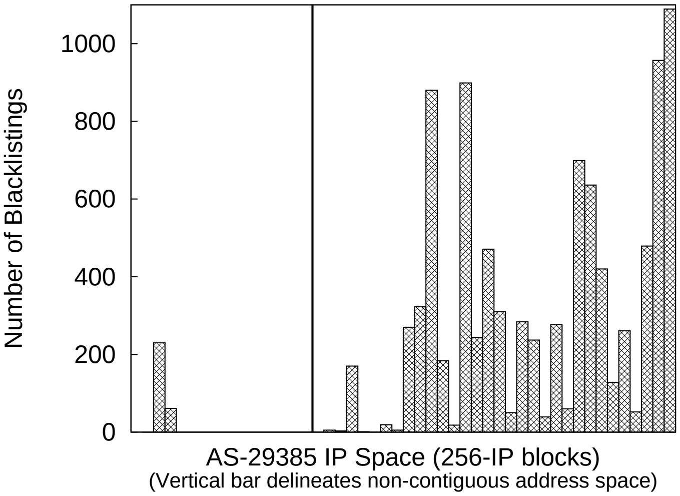

Figure 2: Behavioral Variance within an AS

By adjusting the $decay()$ function in this way, reputations' of SBL IPs are based solely on spatial properties. Our reputation model allows for such flexibility when the feedback database permits. In a similar way, one can focus solely on temporal properties. No matter the case, all such reputations (of varying groupings and decays) can be used in combination (see Sec. 5.7).

4.3 Spatial Groupings

The hierarchical nature of IP address assignment provides natural spatial groupings for use by PRESTA. Starting at the lowest level, a local router or DHCP service assigns IP addresses to individual machines. The selection pool is likely, but not necessarily, well bounded to a subnet (*i.e.*, a $/24$ or $/16$). In turn, these routers operate within an ISP/AS, and ISP/AS get their allocations from Regional Internet Registries (RIRs), whose space is delegated from the Internet Assigned Number Authority [4] (IANA). A clear hierarchy exists, and at each level, a unique reputation can be applied.

We focus our groupings at three levels when considering an IP address: (1) the IP addresses itself, (2) the 768-IP block membership (approximating the subnet), and (3) the AS that homes/originates the IP. It may be the case that an IP address is multi-homed – advertised by two or more ASes. In such situations, where multiple groupings occur for a single grouping function, a data-specific choice should be made. We consider only the highest reputation among the advertising ASes, a choice we discuss further in Sec. 5.5.

In addition to the address space hierarchy, other spatial relationships could be leveraged to form groupings. The social network of email receivers has been proposed as an effective means of spam filtering, as well as network distance [11, 13]. In [15], geographic distance was used as a classifier. Although not investigated herein, groups based on social, routing, or geographic distance could be reasonable, and the addition of these features is an area of future research.

Despite its easily partitioned nature, it has yet to be shown that the IP assignment hierarchy provides relevant groupings. Previous work and anecdotal evidence suggest that AS-number is one of the strongest identifiers of spammers. Indeed, entire AS/ISPs, such as McColo [21] and 3FN [22], have been shut down as a result of their malicious nature. Moreover, in [15], AS-level identifiers were used as a reliable indicator of spamming hosts, after observation indicated only 20 ASes host 42% of spamming IPs.

At the subnet level, we found that groupings of 768 IP-addresses (*i.e.*, three adjacent $/24$s) well contain malicious activity (see Sec. 5.5 for details). As seen in Fig. 2, we visualized $/24$ blocks of address space for

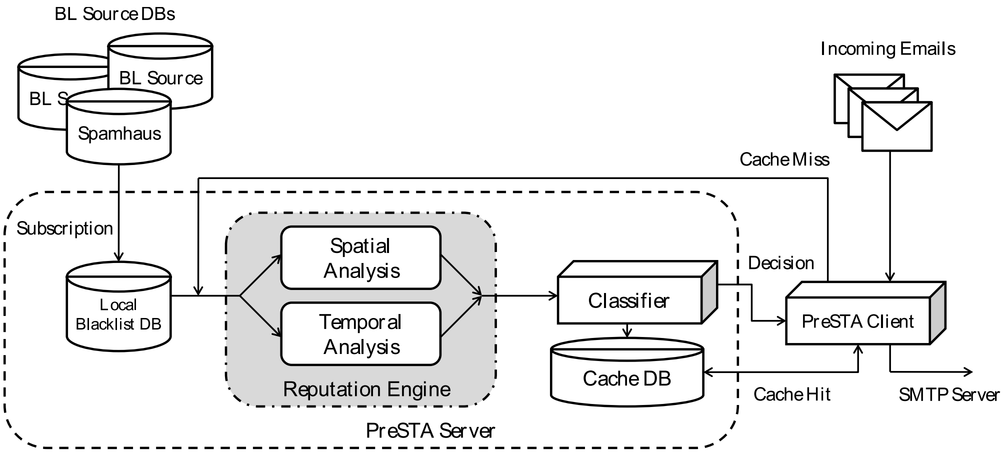

Figure 3: PreSTA **Spam Detection Architecture**

an ISP in Uzbekistan by their quantity of XBL listings. Clearly, there is strong variance across the address space – some regions are highly listed while others have no listings whatsoever. The AS-level reputation of this ISP is comparatively poor due to the quantity of listings, but within the address space, certain block-level reputations are ideal. This suggests that using AS-level reputation alone may be inappropriate because it is too broad a metric. Alternatively, it could be that there is no mail originating from these blocks, but as described in Sec. 5.7, we have automated ways of dealing with such contradictory information.

Finally, using a grouping function that singularly groups entities effectively removes spatial relevance from reputation computation. Intuitively, the reputation of a single IP address should be considered because many mail servers use static addresses. However, the often dynamic nature of address assignment implies that unique IP addresses are not singular groupings, but rather, could represent many different machines over time. A recent study reported that the percentage of dynamically assigned IP addresses[4] on the Internet is substantial and that 96% of mail servers using dynamic IPs send spam almost exclusively [33].

5 Spam Detection Implementation

In this section we describe the implementation of PreSTA for spam detection. The implementation is designed with three primary goals. It should produce a classifier that is (1) lightweight, (2) capable of detecting a large quantity of spam, and (3) do so with a low false-positive rate. We justify our design decisions with respect to these goals. Further, we discuss the practical concerns of such an implementation.

At a high-level, our work-flow begins when an email is received and the connecting IP address and time-stamp of receipt is recorded. Assuming the IP is not actively blacklisted, our spatio-temporal approach is brought to bear. The IP is mapped to its respective spatial groupings: itself, its subnet, and its originating AS(es). Reputations are calculated at each granularity, and these component reputations are supplied as input to a machine-learning classifier trained over previous email. The output is a binary ham/spam label along with each of the three component reputations – all of which may be of use to a client application. We now describe this procedure in detail, and present a visual reference of the PreSTA work-flow in Fig. 3.

[4] Recall that Spamhaus' PBL blacklist is essentially a listing of dynamic IP addresses. We note it is constructed mainly using ISP-provided data, and as such, is far from a complete listing.

5.1 Traditional Blacklists

In Sec. 4.1 the various Spamhaus blacklists were introduced. They not only provide the basis on which reputations are built, but in an implementation of PRESTA, it is natural to apply them as intended – to label emails originating from *currently listed* IPs as spam. When applied to the email data-set, the Spamhaus blacklists (PBL included) captured 91.0% of spam with a 0.74% false-positive rate. Interestingly, this is somewhat higher than previous published statistics[5] [19]. Had we chosen not to exclude the intra-network emails from analysis, the blacklists would have captured a similar 90.9% of spam emails with a much-reduced 0.46% false-positive rate. The exclusion of such emails, while inflating false-positive rates, allows us to concentrate only on the more interesting set of externally-received emails and not bias our results.

The usage of blacklists (independent of spatio-temporal properties), enables fast detection of a large portion of spam emails with minimal time and space requirements. In the implementation of PRESTA, we cache the current blacklists to improve look-up speed, requiring roughly 100MB of space; a reasonable requirement for most email servers. Further, a local store of the blacklist is needed for PRESTA calculation because forms the basis of the historical database.

5.2 Historical Database

Before reputation can be calculated, a historical feedback database must be in place. As described, we retrieve the Spamhaus blacklists at 30-minute intervals. The `diff` is calculated between consecutive copies and time-stamped entries/exits are written to a database. When a new listing appears, we *permanently* record the spatial groups (IP, subnet, and AS(es)) that IP is a member of. For example, if IP i was blacklisted as a member of AS a, that entry will always be a part of a's blacklist history.

We found that roughly 1GB of space is sufficient to store one month's blacklist history (the XBL has 1.0–1.5 million IPs turn over on a daily basis). Fortunately, an extensive history is not required given the exponential $decay()$ function[6]. For example, given a 10-day half-life, a 3-month old XBL entry contributes 0.6% the weight of an active listing. Lengthy histories offer diminishing returns. To save space, one should discard records incapable of contributing statistical significance. Further, such removal saves time because the smaller the set $hist()$ returns the fewer values which must be processed by $raw\_rep()$.

5.3 Grouping Functions

Given an entity (IP address) for which to calculate reputation, we must determine to which groups this entity belongs through the use of our three grouping functions:

- IP FUNCTION: An IP is a group in and of itself, so such a grouping function mirrors its input. As noted earlier, singular groups are interesting because over time, an IP may have multiple inhabitants.
- SUBNET FUNCTION: IP subnet boundaries are not publicly available. Instead, an estimate considers blocks of IP addresses (we use the terms "subnet-level" and "block-level" interchangeably). IP space is partitioned into $/24$s (256 IP segments), and an IP's block grouping consists of the segment in which it resides as well as the segment on either side; 768 addresses per block. Thus, block groupings overlap in the address space, and a single IP input returns one block of IPs (three $/24$s). Although these subnet estimations may overflow known AS boundaries, these naïve blocks prove effective.
- AS FUNCTION: Mapping an IP to its parent AS(es) requires CAIDA [2] and RouteViews [8] data. We note that some AS boundaries overlap in address space and some portions of that space (*i.e.*, unallocated portions) have no resident AS whatsoever. An IP can be homed by any number of ASes,

[5] Our analysis of blacklist performance is from a single-perspective, and therefore may not speak to global blacklist effectiveness.

[6] This minimal history requirement was of benefit to our own study. Reputations must *warm-up* before their use is appropriate. Indeed, our collection of blacklist data began in 5/2009, three months before our first classifications.

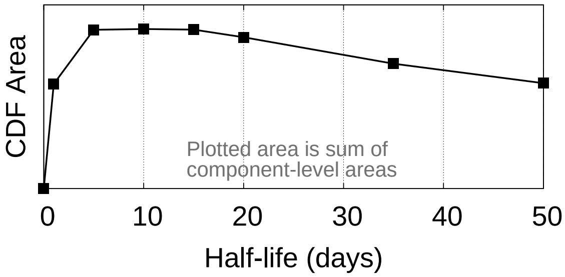

Figure 4: Affect of Half-Life on CDF Area

including none at all, the technical considerations of which are addressed in Sec. 5.5. The function's output is all the IPs homed by an AS(es) in which the input IP is a member. Each returned IP is tagged with the parent AS, so a well-defined subset of the output can be chosen.

5.4 Decay Function

The decay function controls the extent to which temporal proximity factors into reputation. This is configured via the half-life parameter, h. If h is too small, reputations will decay rapidly and provide little benefit over using blacklists alone. Too large an h can cause an increase in false positives due to stale information.

A good half-life should maximize the difference between the reputation of spam and ham email, and to arrive at a reasonable value for h, we analyzed a set of emails/reputations pre-dating our evaluation period. By plotting the reputation-CDF for both spam and ham email, we sought a value for h that maximized the area between the curves. In Fig. 4 we present the calculations from these experiments. We found $h = 10$ (days) to optimal and therefore use this value in our spam application[7]. With the half-life established, and having previously chosen $d = 5$ (days), we calculate MAX_REP$= 4.14$.

As described previously, we actually employ two separate $decay()$ functions depending on where a listing appeared, either on the SBL or the XBL. Manually maintained, we do not decay de-listing for the SBL, but the XBL is decayed using the aforementioned 10-day half-life. In order to use both listings in combination, we apply a flag to each time pair returned by $hist()$ dependent on which blacklist they originated from, allowing us to apply the appropriate decay function.

5.5 Reputation Calculation

Given the decay function (Sec. 5.4), output (sets of IP addresses) of the three grouping functions (Sec. 5.3), and the feedback database (Sec. 5.2), reputation may now be calculated. Valuation is performed at each granularity; three reputation values are returned. Calculation closely follows as described in Sec. 3.

Calculation of IP-level and subnet-level reputation is straightforward per the reputation model with $size() = 1$ and $size() = 768$, respectively. The particulars of AS-level calculation are more interesting. An IP may be a member of any quantity of ASes, including none at all. If an IP is multi-homed, we make the conservative choice by selecting the most reputable AS value as the AS-level reputation. Those IPs mapping to no AS form their own group, and we designate the reputation for this group as 0 because, in general, unallocated space is only used for malicious activity. In this spatial grouping, $size()$ is not constant over time. Instead, magnitudes are pre-computed for all AS using CAIDA data and updated as BGP routes change. Only *unique* originating IPs are considered (blocks often overlap to support traffic engineering).

[7] Although we found it unnecessary, h could be optimized on an interval basis, much like re-training a classifier. However, our experiments showed minor variations of the parameter to be inconsequential.

5.6 Calculation Optimizations

To be lightweight, our system must calculate reputation efficiently. It should not significantly slow email delivery (latency), and it should handle heavy email loads (bandwidth). We now describe caching strategies and other techniques in support of these goals:

- AS VALUE CACHING: Reputations for *all* ASes are periodically recalculated off-line. These calculations are (relatively) slow given their $hist()$ calls return large sets.
- BLOCK/IP VALUE CACHING: Similarly, block and IP reputations can be cached after the first cache miss. Cache hit rates are expected to be high because (1) an email with multiple recipients (*i.e.,* a carbon copy) is received multiple times but with the same source IP address, and (2) source IP addresses are non-uniformly distributed. For the 6.1 million (non-Penn, non-blacklisted) emails in our working data-set, there are 364k unique IP senders and 176k unique sender 'blocks.'
- CACHE CONSISTENCY: Caches at all levels need to be cleared when the blacklists are updated (every 30 minutes), to avoid inconsistencies involving the arrival of new listings. As far as time-decay is concerned, a discrepancy of up to 30 minutes is inconsequential when considering a 10-day half-life.
- WHITELISTING: There is no reason to calculate reputation in trusted IP addresses, such as one's own server. Of course, whitelists could also be utilized in a feedback loop to alleviate false-positives stemming from those entities whose emails are consistently misclassified.

Using these optimizations, our PRESTA implementation is capable of scoring 500k emails an hour, with average email latencies well under a second. Latency and bandwidth are minimal concerns. Instead, it is the off-line processing supporting this scoring which is the biggest resource consumer. Even so, our implementation is comfortably handled by a commodity machine and could easily run adjacent to an email server. Pertinent implementation statistics, such as cache performance, are available in Sec. 6.4.

5.7 Reputation Classification

Extraction of a binary classification (*i.e.,* spam or ham) is based on a *threshold* strategy. Emails valuated above the threshold are considered ham, and those below are considered spam. Finding an appropriate threshold can be difficult, especially as dimensionality grows, as is the case when classifying multiple reputation values. Further, a fixed threshold is insufficient due to temporal fluctuations; as large groups (botnets) of spamming IPs arise and fall over time, and the distinction between good and bad may shift.

Support vector machines (SVM) [17] are employed to determine thresholds. SVM is a form of supervised learning that provides a simple and effective means to classify multiple reputation values. The algorithm maps reputation triples (a feature for each spatial dimension) from an email training set into 3-dimensional space. It then determines the surface (threshold) that best divides spam and ham data-points based on the training labels. This same threshold can then be applied during classification. The SVM routine can be tuned via a *cost* metric which is correlated to the eventual false-positive rate of the classifier.

The classifier is adjusted (re-trained) every 4 days to handle dynamism. A subset of emails received in the previous 4 days are trained upon, and the resulting classifier is used for the next 4 day interval. The affect of different training periods has not been extensively studied. Clearly, large periods are not desired; the reputation of distant emails may not speak to the classification of more current ones. Too short a period is equally poor because it requires extensive resources to re-train so frequently. We believe 4-day re-training is a good compromise. However, the re-training period need not be fixed, and future work will explore re-training rates that adjust based on various environmental factors at the email server.

At each re-training, we used 10,000 emails (5% of the non-Penn, non-blacklisted email received every 4 days), and labeled emails as spam/ham based on the Proofpoint score. In a more general use case, there would be some form of client feedback correlated across many accounts that can classify spam post-delivery

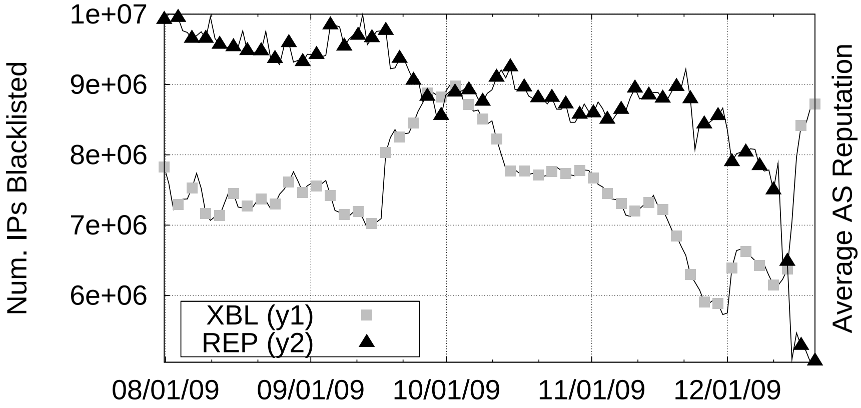

Figure 5: XBL Size Relative to Global Reputation

and train various spam detectors. Since we do not have access to such user behavior, correlation statistics, or any external spam filters, the provided Proofpoint values are assumed.

Post-training, the false-positive (FP) rate of the classifier can be estimated by measuring the error over the training set (assuming one does not over-fit the training data). The estimated FP-rate is a good indicator of the true FP-rate, and the SVM cost parameter can be adjusted to tune the expected FP-rate. All classifier statistics and graphs hereafter were produced with a 0.5% tolerance for false-positives (over the classification set), as this simplifies presentation. We believe 0.5% is a reasonable setting given that blacklists are widely accepted and achieved a 0.74% FP-rate over the same dataset. Additionally, these rates are somewhat inflated given our decision to exclude intra-network emails, which are unlikely to contribute false-positives (the blacklist FP-rate reduced to 0.46% with their inclusion). In Sec. 6.5, the trade-off between the FP-rate and spam blockage is examined in greater depth.

6 Empirical Results

We begin our PRESTA spam detection analysis by examining the component reputations individually. From there, two case studies will exemplify how PRESTA can produce metrics outperforming traditional blacklists in both spatial and temporal dimensions. Finally, we examine the effectiveness of the full-fledged spam filter: For each mail in our data-set, reputation metrics are calculated and email is valuated in a way that would mimic our PRESTA implementation on a production email server, complete with re-training and caching.

6.1 Blacklist Relationship

In examining how our reputations quantify behavior, we began with a simple intuition: one would expect to see a clear push-pull relationship between an entities reputation and the number of corresponding entries on the blacklist. To confirm, we graphed the size of the XBL blacklist[8] over time and compared this to the average reputation of *all* ASes. Results are presented in Fig. 5. We observe an inverse relationship, confirming our expectations. When the number of listings dips, reputation increases – and vice versa.

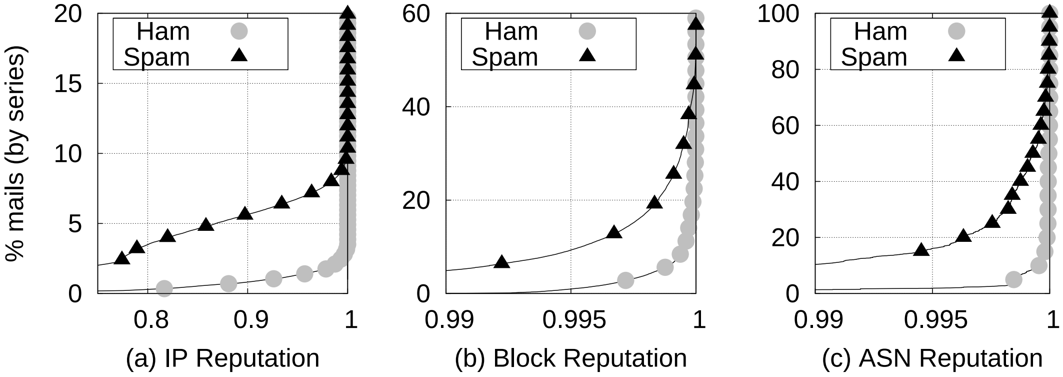

Figure 6: CDFs of Component Reputations

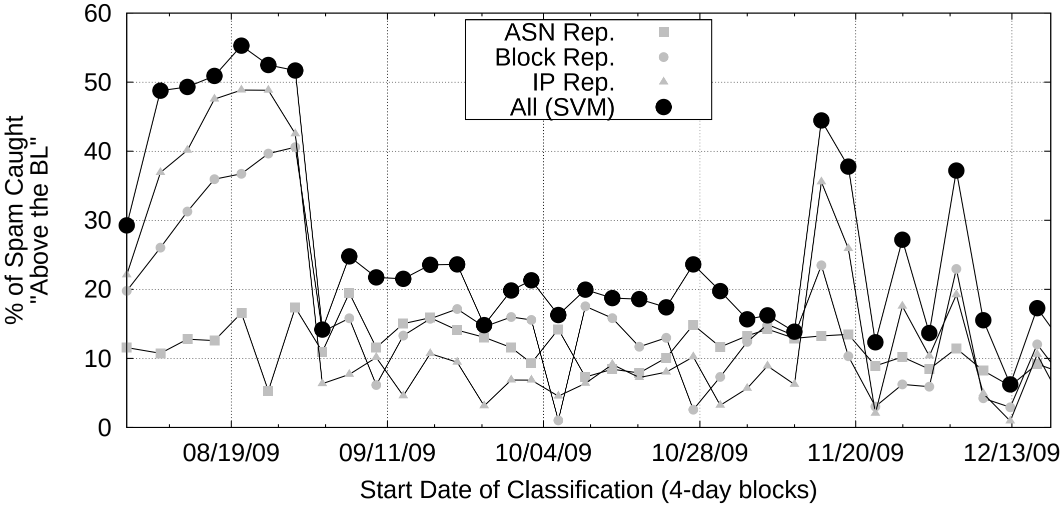

Figure 7: Contribution of Component Reputations

6.2 Component Reputation Analysis

In order for component reputations (IP, block, and AS) to be useful in spam detection they must be *behavior predictive*. That is, the reputations associated to ham emails should exceed those of spam emails. This relationship is visualized in the cumulative-distribution-functions (CDFs) of Fig. 6. We observe that all component reputations behave as expected. Fig. 6 also shows the benefit of using multiple spatial groupings. While nearly 90% of spam emails come from IPs that had ideal reputation (*i.e.*, a reputation of 1) at the time of receipt, this is true for just 46% of blocks, and only 3% of AS.

The CDFs of Fig. 6 imply that each component reputation is, in and of itself, a metric capable of classifying some quantity of spam. However, it is desirable to show that each granularity captures *unique* spam, so that the combination of multiple reputations can produce a higher-order classifier of greater accuracy. In Fig. 7, the effectiveness of each component reputation is presented. The percentage of spam caught is “above the blacklist,” or more precisely, the percentage of spam well-classified by the reputation value that was not identified by the blacklist alone. Given the inclusion of traditional blacklist filtering, the primary concern is those emails that are not actively listed.

On the average, PRESTA is able to capture 25.7% of spam emails not caught by traditional blacklists.

[8] The XBL is the driving force behind reputation. The SBL is also a contributor, but is orders of magnitude smaller.

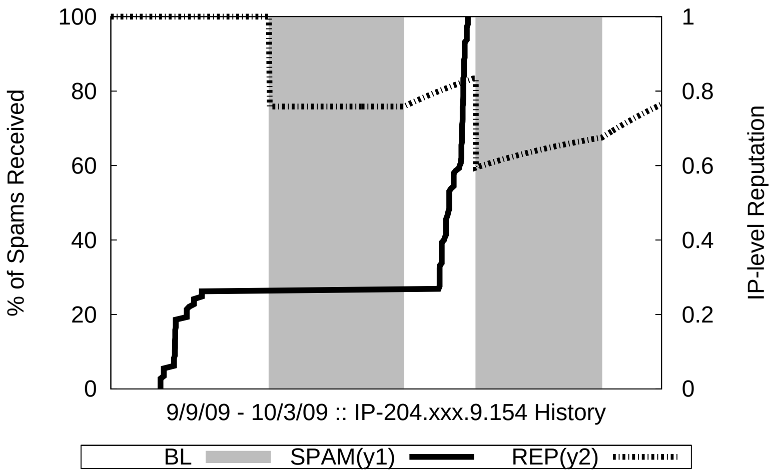

Figure 8: Single IP Behavior w.r.t. Blacklisting

Crucially, the combined performance (the top line of Fig. 7), exceeds that of any individual component, so each spatial grouping catches spam the others do not.

We are also interested in determining which grouping provides the best classification. AS-level reputation is the most stable of the components, individually capable of classifying an additional 10-15% of spam above the blacklist. However, we observe that during periods of increased PreSTA performance, it is often the block and IP levels that make significant contributions. This is intuitive; AS-level thresholding must be conservative. Given their large size, ASes have relatively stable reputations. Thus, the classification of a single reputation value may effectively make the spam/ham determination for many thousand emails – and could result in an unacceptable increase in the FP-rate. Meanwhile, the cost associated with a mis-prediction is far less for block and IP groupings, permitting more aggressive/speculative thresholds.

These results suggest that considering more spatial dimensions can increase performance, that is, when there are non-overlapping classifications. However, there are diminishing returns. Each additional component reputation requires increased resources for valuation and classification. An application should seek a minimal set of dimensions to best represent and classify its data.

6.3 Case Studies

Two case studies are exemplary of the types of spam behavior able to evade blacklists, yet captured via PreSTA. First, Fig. 8 shows the *temporal* sending patterns of a single spamming IP address. This IP was blacklisted twice during the course of the study (as indicated by shaded regions), and the time between listings was small (roughly 2 days). The controller of this IP address likely used blacklist counter-intelligence [26] to increase the likelihood that spam would be delivered. Notice that no spam was observed when the IP was actively blacklisted, but 150 spam emails were received at other times.

Traditional blacklist are reactive, binary measures that do not take history into account. During the intermittent period between listings, as far as the blacklist is concerned, the spamming IP is an innocent one. However, as shown in Fig. 8, the IP-level reputation metric compounds prior evidence. Thus, if PreSTA had been in use, the intermittent influx of email would have been identified as spam.

Secondly, Fig. 9 visualizes a case study at the AS-level utilizing both *spatial* and *temporal* dimensions. In the early stages of our data collection we noticed anomalous activity occurring at a particular AS (AS#-12743)[9]. Even when compared to the other four worst performing ASes during the time block, ASN-12743's

[9] PTK-Centertel, a major Polish mobile service provider

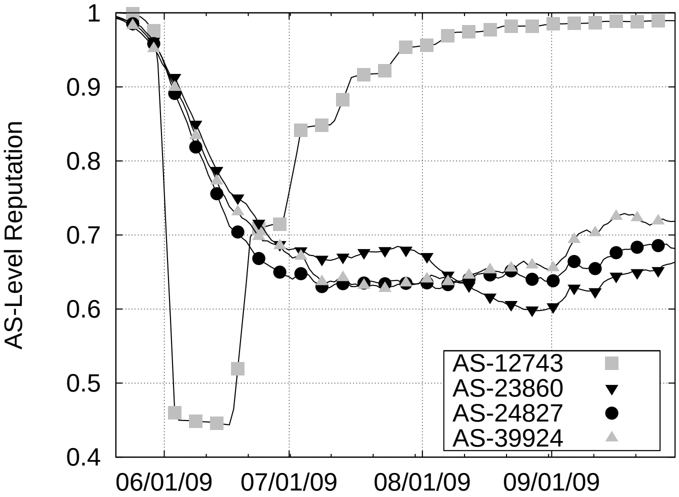

Figure 9: Temporal Shift within Spatial Grouping (AS)

drop in reputation is astounding. Nearly its entire address space, some 4,500 addresses, were blacklisted in the course of several days – likely indicative of an aggressive botnet-based spam campaign – after which, the reputation increases exponentially (per the half-life), eventually returning to innocent levels.

With traditional blacklists, an IP must actually send spam before it can be blacklisted. In the ASN-12743 case, this means all 4,500 IPs had some window in which to freely send spam. However, as the IPs were listed in mass, the *reputation* of the AS drops at an alarming rate, losing more than 50% of its value. Had PreSTA been implemented, the reputation of the AS (and the blocks within) would have been low enough to classify mails sourced from the remainder of the space as spam, mitigating the brunt of the attack.

6.4 Implementation Performance

The previous case studies are but two examples of the way component reputations capture spamming behavior that evades traditional blacklists. We now present the results of the simulated PreSTA implementation.

Our experimental setup was as follows: To best simulate the normal processing at a mail server, we assumed each email arrived in the order of the time-stamp. The blacklist history and cached reputation scores were regulated so that only the knowledge available at the time of arrival is used to valuate the email. PreSTA requires a warm-up period to gather enough temporal knowledge to process correctly; hence, historical blacklist storage began three months prior to the first email being scored.

We were interested in measuring the effectiveness of the cache and the latency of the system. Caching was highly effective: 56.8% of block-level calculations are avoided, and 43.1% are avoided at the IP-level (recall that *all* AS-level calculations are performed off-line and then cached). As such, the reputation of an incoming email can be calculated in nearly real time, with the average email being processed in fractions of a second. Under typical conditions, over 500,000 emails can be scored in an hour.

Re-training our classifiers and rebuilding the AS-cache are the most time consumptive activities. Fortunately, finding new classifiers takes only minutes of work for a 10,000 email training set, and only needs to be performed every 4 days. Re-training can also be done off-line and not affect current scoring. Rebuilding the AS reputation cache must be done every 30 minutes once new blacklist data is available, but it need not delay current scoring as the previous AS-level reputations are still relevant (at most 30 minutes old).

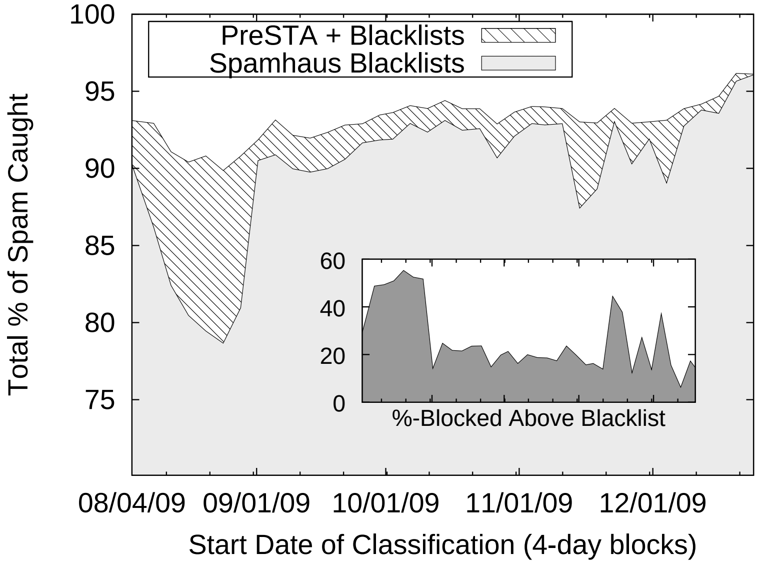

Figure 10: Blacklist and PRESTA Spam Blockage

6.5 Spam Mitigation Performance

The spam detection capabilities of PRESTA are summarized in Fig. 10. On average, 93% of spam emails are identified when using our system in conjunction with traditional blacklists. To some this may seem to be only a nominal increase over using blacklists alone. However, the inset of Fig. 10 is more intuitive; plotting the PRESTA blockage rate only over those mails passing the Spamhaus blacklists (identical to the top line seen in Fig. 7). We observe that between 20% and 50% of spams evading blacklists can be caught by PRESTA (with a 25.7% average). Had PRESTA been implemented on our university mail server, it would have caught 650,000 spam emails that evaded the Spamhaus blacklists over the course of our study.

Most interestingly, PRESTA allows for a consistent and steady state of spam detection. For example, consider the significant drops in blacklist performance seen throughout our study (for example, in late August 2009 and again in mid-November 2009). PRESTA is nearly unaffected during these periods and could be used as a stop-gap to variance in blacklist accuracy. Clearly, whatever the means of blacklist evasion was during these periods, it was insufficient to evade PRESTA. Further, we believe future data will show such dips and rises in blacklist performance to be non-anomalous. Periods of high de-listing are likely to be followed by periods of high re-listing as spammers try to maximize the utility of available IPs. In the interim, blacklists are likely to perform relatively poorly, and PRESTA could aid in maintaining a consistent level of spam blockage. While the blockage-rates of the blacklists fluctuate 18% over the course of our study, PRESTA is far more consistent, exhibiting just 5% of variance.

Ultimately, the performance attainable by our classifier is dependent on the number of false-positives (FPs) a user is willing to tolerate. To this point, the FP-rate has been fixed at 0.5% in order to simplify discussion. However, as exemplified in Fig. 11, the FP-rate is tune-able and strongly correlates with the blockage rate. The plot is generated over a characteristic interval of email from mid-October 2009, and is akin to the precision/recall graphs common in machine-learning. We remind readers that our decision to exclude intra-network emails from our dataset (see Sec. 5.1) significantly inflates the presented FP-rates.

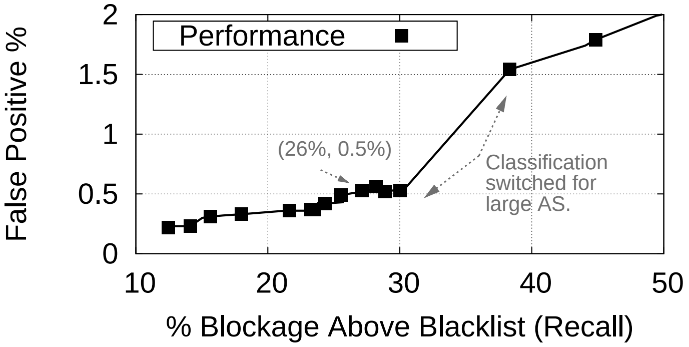

Figure 11: Characteristic FP/Blockage Trade-Off

7 Evasion and Gamesmanship

In order for a PRESTA-based spam filtering approach to be effective, it must be robust to evasion. Given that we use blacklists as a feedback source, perhaps the most effective way to avoid PRESTA detection is to avoid getting blacklisted in the first place (this is true in any PRESTA application where one can avoid negative feedback). However, such a technique is not fail-safe; a single evasive entity may still have poor reputation at broader granularity. Given that negative feedback does exist, and an IP has been blacklisted, the best recourse is patience. Over time, the weight of the listing will decrease according to the decay function. As such, there is no way to evade PRESTA in the temporal dimension.

However, spammers are migrant, and the spatial dimension affords greater opportunities. While IP and block magnitudes are fixed, an AS controls the number of IPs it broadcasts. An actively evasive AS would ensure its entire allocation is broadcasted. More maliciously, a spammer may briefly hijack IP space they were *not* allocated in order to send spam from stolen IPs. Such *spectrum agility* was shown by [25] to be an emergent spamming technique. Fortunately, if the hijacked IP space was not being broadcasted (*i.e.*, unallocated), emails from these IPs would map to the special grouping “no AS”, whose reputation is zero (per Sec. 5.5). However, if the hijacked space was being broadcasted by a reputable AS, evasion may be possible. Fortunately, [25] observes the use of unallocated space is most prevalent.

As a general purpose reputation engine, a *sizing attack* can be of real concern. The entities being valuated should not be able to affect the size of their spatial groupings. However, this attack is only effective when the group size is non-singular, and an easy avoidance technique is to always include a grouping function that defines singular groups. Further, an implementation should try to assign persistent identifiers to entities. When identifiers are non-persistent, PRESTA could fall victim to a Sybil attack [12] since an entity could evade negative feedback by simply changing identifiers.

8 Additional PRESTA Applications

PRESTA’s applicability is broader than email spam alone, as the spatial and temporal properties described are inherent in a number of domains. Indeed, PRESTA reputation values have already proven successful in the detection of vandalism on Wikipedia [32]. Any edit which is blatantly unproductive, offensive, or over-zealous in its removal of content is said to exhibit *vandalism*. Prior to [32], attempts to detect these edits resided only in the language-processing domain [23, 29]. Akin to the Bayesian filters of email spam, these efforts also suffer from many of the same drawbacks (evade-ability and minimal throughput).

Recall our two PRESTA application criteria. First, A view-able history of dynamic negative feedback. On Wikipedia, a special administration form of reversion called *rollback* permits the discovery of malicious edits, fulfilling this requirement. Second, there should be at least one finite partitioning of the entities. The authors of [32] find it appropriate to consider both *users* and *articles* as the entities involved in an edit. In addition to singular groupings, these entities are grouped by geographical-space and topic-space (*i.e.,* categories), respectively. Combining PreSTA values with other metadata features, [32] ultimately produces a classifier comparable in performance to natural-language efforts.

Other use-cases for PreSTA are active areas of research, and in particular, content-based access control scenarios seem most ripe for exploration. However, it may be possible to generalize the PreSTA model further by providing a reduction to *dynamic trust management* (DTM) systems [10, 31], which combine trust management and reputation management fundamentals to make access control decisions using only partial information. Credential delegation chains have hierarchical properties from which spatial groups can be extracted. Moreover, DTM systems rely on feedback databases for their reputation component, which PreSTA could easily leverage. Future work could formalize this reduction, showing that PreSTA may be applicable to an entire new class of systems.

9 Conclusions

In this paper, we have introduced PreSTA, a spatio-temporal reputation model, and demonstrated its effectiveness by using it for spam detection. PreSTA has proven capable with respect to spam, blocking up to 50% of spam emails not caught by traditional blacklists, and identifying 93% of spam on average when used in combination. In particular, our method succeeded in being a stop-gap mechanism for periods of low blacklist performance. Our technique is also scalable and able to efficiently handle production email workload, at least at the level of a university mail system, processing over 500,000 emails an hour.

We do not propose PreSTA based spam detection as a replacement for context-based analysis systems. However, we believe it could be useful as an intermediate filter, perhaps centrally maintained and queried like DNS-based IP blacklists are today. Alternatively, the reputation values PreSTA computes could be used in combination with other features to produce classifiers of even greater accuracy. With only a small amount of extra processing, PreSTA was able to turn a reactive blacklist into a predictive service.

Further, we believe PreSTA has applicability beyond spam, and recent related work with Wikipedia has already shown this to be the case. Any application meeting our two criteria is a potential use case. No matter the application, PreSTA's power is derived from its ability to take a historical record of bad behavior and produce from that predictive identifications of *additional* malicious entities. This is achieved by analyzing not just an individual entity's historical behavior, but also the histories of groups wherein the entity resides. Thus, in the absence of entity-specific data, we are able to rely on spatial and temporal data to make a characterization. By combining reputations from varying granularity, we are able to produce robust reputation values that, as a result of normalization, are comparable. Ultimately, the reputations may be utilized as an effective means of performing dynamic access-control and mitigating malicious behavior, two extremely relevant issues as service paradigms shift to more distributed architectures.

References

[1] Apache SpamAssassin Project. `http://spamassassin.apache.org/`.

[2] CAIDA - The Cooperative Association for Internet Data Analysis. `http://www.caida.org/`.

[3] DNSBL.info - Spam database lookup: Blacklists. `http://www.dnsbl.info/dnsbl-list.php`.

[4] IANA - Internet Assigned Numbers Authority. `http://www.iana.org/`.

[5] MessageLabs Intelligence reports. `http://www.messagelabs.com/intelligence.aspx`.

[6] Proofpoint, Inc. `http://www.proofpoint.com/`.

[7] Spamhaus Project. `http://www.spamhaus.org/`.

[8] University of Oregon Route Views Project. `http://www.routeviews.org/`.

[9] D. Alperovitch, P. Judge, and S. Krasser. Taxonomy of email reputation systems. In *Distributed Computing Systems Workshops - ICDCSW '07*, June 2007.

[10] M. Blaze, S. Kannan, A. D. Keromytis, I. Lee, W. Lee, O. Sokolsky, and J. M. Smith. Dynamic trust management. *IEEE Computer (Special Issue on Trust Mangement)*, 2009.

[11] P. Boykins and B. Roychowdhury. Personal email networks: An effective anti-spam tool. In *IEEE Computer 38*, volume 38, pages 61–68, 2005.

[12] J. Douceur. The Sybil attack. In *1st IPTPS*, March 2002.

[13] J. Golbeck and J. Hendler. Reputation network analysis for email filtering. In *In Proceedings of the Conference on Email and Anti-Spam (CEAS)*, 2004.

[14] J. Goodman. IP addresses in email clients. In *Proc. of the Conference on Email and Anti-Spam (CEAS)*, 2004.

[15] S. Hao, N. A. Syed, N. Feamster, A. G. Gray, and S. Krasser. Detecting spammers with SNARE: Spatio-temporal network-level automated reputation engine. In *18th USENIX Security Symposium*, August 2009.

[16] IronPort Systems Inc. Reputation-based mail flow control. White Paper, 2002. (SenderBase).

[17] T. Joachims. *Advances in Kernel Methods - Support Vector Learning*, chapter Making Large-scale SVM Learning Practical, pages 169–184. MIT Press, Cambridge, MA, 1999.

[18] A. Jøsang, R. Hayward, and S. Pope. Trust network analysis with subjective logic. In *Proceedings of the 29th Australasian Computer Science Conference*, 2006.

[19] J. Jung and E. Sit. An empirical study of spam traffic and the use of DNS black lists. In *IMC '04: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, pages 370–375, 2004.

[20] S. D. Kamvar, M. T. Schlosser, and H. Garcia-molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the Twelfth International World Wide Web Conference*, Budapest, May 2003.

[21] B. Krebs. Host of Internet spam groups is cut off. `http://www.washingtonpost.com/wp-dyn/content/article/2008/11/12/AR2008111200658.html`, November 2008. (McColo Corporation).

[22] B. Krebs. FTC sues, shuts down N. Calif. web hosting firm. `http://voices.washingtonpost.com/securityfix/2009/06/ftc_sues_shuts_down_n_calif_we.html`, June 2009. (3FN).

[23] M. Potthast, B. Stein, and R. Gerling. Automatic vandalism detection in Wikipedia. pages 663–668, 2008.

[24] A. Ramachandran, D. Dagon, and N. Feamster. Can DNSBLs keep up with bots? In *3rd Conference on Email and Anti-Spam (CEAS)*, 2006.

[25] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *Proceedings of SIGCOMM 2006*, pages 291–302, 2006.

[26] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using DNSBL counter-intelligence. In *USENIX: Steps to Reducing Unwanted Traffic on the Internet*, pages 49–54, 2006.

[27] A. Ramachandran, N. Feamster, and S. Vempala. Filtering spam with behavioral blacklisting. In *Proc. of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pages 342–351, 2007.

[28] M. Sahami, S. Dumais, D. Heckerman, and E.Horvitz. A Bayesian approach to filtering junk e-mail. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[29] K. Smets, B. Goethals, and B. Verdonk. Automatic vandalism detection in Wikipedia: Towards a machine learning approach. In *WikiAI '08: Proc. of the AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.

[30] Symantec Corp. IP reputation investigation. `http://ipremoval.sms.symantec.com/`.

[31] A. G. West, A. J. Aviv, J. Chang, V. S. Prabhu, M. Blaze, S. Kannan, I. Lee, J. M. Smith, and O. Sokolsky. QuanTM: A quantitative trust management system. In *EUROSEC 2009*, pages 28–35, March 2009.

[32] A. G. West and I. Lee. Detecting Wikipedia vandalism via spatio-temporal analysis of revision metadata. Technical Report MS-CIS-10-05, University of Pennsylvania, February 2010.

[33] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How dynamic are IP addresses? In *In Proceedings of SIGCOMM '07*, 2007.